SRA Quality Scoring Specification

Version 0.2 Draft B 27 Sep 2009

National Center for Biotechnology Information – National Library of Medicine
European Bioinformatics Institute
Sanger Institute – Wellcome Trust

Contents

Overview

Since 2007 the sequence read archives (SRA) at NCBI, EBI, and DDBJ have been collecting raw sequencing data from a variety of "next generation" sequencing platforms: Illumina, SOLiD, 454, Helicos, CompleteGenomics, and others. During a long period of introduction, deployment, and update, these platforms are now producing data in vast quantities. The Archives have been collecting raw data in many forms with the minimal information consisting of individual base/color calls and quality scores for those calls. Additional data has been accepted including various levels of intensity measurements and scores. Naturally this has led to an order of magnitude increase in the cost of archiving. However, the minimal information content is not always sufficient to capture all the error events that may occur in sequencing, and which may be important in certain bioinformatics such as variation detection.

In order to establish a more uniform definition of both minimal and maximal information content for raw sequencing data deposition, the Archives have proposed in this document a method for defining, measuring, and archiving quality scores for raw sequencing data. Having such a standard will allow quality scores from various platforms to be compared while also allowing for capture of error events that are not universally represented.

This document is an initial proposal that can provide a starting point for discussion and possible future standardization.

Goals

- Define a standard range of quality values.
- Define standard methods for computing quality from error events.
- Define a flexible way to represent platform-specific error events such as indels.
- Define methods for quality score interchange between platforms where appropriate.
- Define an abstract layer so that the need for archiving raw intensity measurements can be eliminated.

Related Documents

Contacts

To comment on this document, please email:

Event-qual-sra mailing list: Event-qual-sra@ebi.ac.uk
http://listserver.ebi.ac.uk/mailman/listinfo/event-qual-sra

Revision History

18 Sep 2009 - Final reviewable version compiled by James Bonfield from draft written by Eugene Yaschenko.

1. Data submission levels.

We define three primary submission levels for the sequence data:

- (1.1) Sequence plus one quality value
- (1.2) Sequence plus multiple quality values
- (1.3) Sequence plus trace intensities and one or more quality values

We are proposing to only use the first 2 levels of submissions as a target of future archiving. During transitional period the third level (current signals) will exist for legacy support.

An important note: the 3 layers above may not necessarily have exactly the same number of reported data points. This will allow insertion-alternates to be represented as an extra data point in 1.2 with respect to 1.1.

1.1 Primary base calling level.

This is the best fully defined representation of reported sequences as stated by submitter/vendor. Existence of every base is defined and quality score in the phred scale is assigned. This level is a minimum threshold for submitting any data to sequence archives. It may also become the maximum for experiments designed to provide "counting level" data (CHiP-Seq, RNA-Seq, etc). This level is sometimes called "1+1" and "fastq".

1.2 Event probability level.

This level is designed to provide alternative events using assigned probabilities. There will be a fixed number of events for a given technology with probabilities assigned to each of them. See section 2 below for a more detailed explanation of this level. Having the ability to represent alternative events (calls, indels) makes this level a necessity for storing "variant level" data (polymorphism, cancer, etc).

1.3 Existing signal levels (454, Illumina, AB SOLiD).

This is the most space-consuming and we will be looking into reducing its usage by introducing 1.2 for many platforms. For some platforms (eg 454) the signals still carry valuable information so this level will not completely vanish.

2. Event probability model.

2.1 Definitions.

Every platform will develop a fixed number of events it is designed to measure. For example "A,T,G,C" for Illumina, "+(base present at a given flow),-(no base)" for 454.

Probability number 0<p<1 will be computed for every event to occur.

Errors may occur through base-calling (eg noisy signals) or in earlier stages such as library preparation (eg PCR errors). This typically leads to a mixture of per-base error rates and per-library error rates. The events primarily accommodate the per-base probabilities.

2.2 Rules.

- There should always be a non-zero probability of an error (none of the events have occurred), meaning that sum(p) < 1.

- It will be allowed to set p=0 for an event. While this is not theoretically possible, the meaning of 0 will be that the current pipeline is not able to distinguish the probability for this event from the probability of a generalized error.

2.3 Numerical treatment.

During computation keeping probabilities as 32-bit floats will provide sufficient level of precision. This is 1+n approach and it is as close to the event model as possible. However, it will be very expensive to archive. The best model for archiving will be to switch to n+n. It will be done the following way:

2.3.1 events will get sorted by most probable: pa, pb, pc, pd, ...

2.3.2 the probabilities will get recomputed as 1-pa, 1-pa-pb, 1-pa-pb-pc, etc.

2.3.3 the order of events will be archived

2.3.4 recomputed numbers from 2.3.2 can either be stored as floats with low-precision mantissa or as phred-like conversion to integers:

```
    -10log(1-pa),
    -10log((1-pa-pb)/(1-pa)),
  -10log((1-pa-pb-pc)/(1-pa-pb))
```

2.3.5 the granularity and/or precision of measurements will vary by instrument and software. We expect the producers of this data to indicate what storage precision is applicable.

3. Application to existing/new technologies

We use the term "clock" here to indicate an instrument that governs the rate at which DNA bases are incorporated. In this way the measurements taken can be synchronised with the DNA chemistry. Conversely "unclocked" instruments may have a regular time period of taking measurements, but they have no bearing on the actual rate of DNA sequencing - an example being the Sanger electrophoresis sequencing method.

3.1 Illumina

All three levels (1.1-1.3) are having the same clock - flow. As a result no indels occur within the measurements. (Indels may occur at library preparation stage.)

We define four events: A, T, G, C.

Note that the existing uncalibrated 4-channel quality score is not precise enough (possibly truncated) to restore event probabilities. Multiple efforts were done to restore probabilities from intensities. See section A.3 for more information on this.

3.2 AB SOLiD

All three levels have the same clock.

We define four events: 1, 2, 3, 4 (color space).

It is possible to translate events 1,2,3,4 into A,T,G,C by matrix algebra, but it is generally not recommended until the last stage of any alignments / calculations.

3.3 454

Clock is based on alternating flows with each flow incorporating a variable number of bases, as a result 1.1 - 1.3 have different number of data points.

We define two events:
'+' (current flow produces a call)
'-' (no call at the current flow).

Multiple data points can be collected at a single flow. Indels: deletes are already recorded as '-', inserts may be created by adding more data points on 1.2 level (overcalling).

3.4 Sanger/PacBio

The clock is time-based, so we expect a different number of data points in 1.3 versus 1.1 and 1.2.

We define events: A,T,G,C.

Inserts can be recorded by adding additional data points at 1.2 level.

3.5 Helicos

Most likely similar to 454, but currently is planning to supply 1+1 only.

3.6 Complete Genomics

Currently is planning to supply 1+1 only.

Appendix A - Format Conversions

While not part of the requirements, this appendix covers the practicalities of using this data.

A.1 Fastq

Fastq requires just one single quality value. Although a certain degree of fragmentation has occurred over the ASCII offset used for encoding the quality values, the conversion is essentially the same.

We need to take the first (most probable) event 'a' and convert this to a probability value.

Probability = 1 - 10**(a/-10)

(Where a**b denotes "a to the power of b")

For overcalls and undercalls, the base should be produced in the fastq output if the base has at least 0.5 probability of existing.

A.2 SAM

The SAM specification currently uses a 2+2 model; the primary called base has a probability value associated with it, but optionally the second most likely base can be stored along with a second probability value.

This is trivially converted from using the first two values in the n+n storage as described in 2.3.

A.3 Raw probability values

The reverse of 2.3.4 above is straight forward. Given 4 events with probability pa, pb, pc, pd we produce 4 encoded event probabilities a, b, c, d:

a = -10log(1-pa)
b = -10log((1-pa-pb)/(1-pa))
c = -10log((1-pa-pb-pc)/(1-pa-pb))
d = -10log((1-pa-pb-pc-pd)/(1-pa-pb-pc))

A worked example of P{a,b,c,d} = {0.8, 0.1, 0.07, 0.02} (0.01 error) gives:

a = 6.99
b = 3.01
c = 5.23
d = 4.77

The reverse simply becomes:

pa =   1 - 10**(a/-10)
pb =   (1-pa) * (1 - 10**(b/-10))
pc =   (1-pa-pb) * (1 - 10**(c/-10))
pd = (1-pa-pb-pc) * (1 - 10**(d/-10))

The rationale of using this encoding mechanism is to spread the impact of loss of precision. The error caused by writing out a=7 instead of a=6.99 is evenly spread across the b, c and d events. Similarly the error in storing b=3.01 is evenly spread between the c and d events.

In contrast the existing four Illumina scores, stored using a log-odds system, give rise to larger errors. The main problem here is inherent in loss of precision causing the four values to not add up to one.